# Growing needs for the Global IT market

Lawrence Chinn

21 Nov 2024

IBM Developer

SKILLS NETWORK

# OUTLINE

1. **Executive Summary**
2. **Introduction**
3. **Methodology**
4. **Results**
   - Visualization – Charts
   - Dashboard
5. **Discussion**
   - Findings & Implications
6. **Conclusion**
7. **Appendix**

# EXECUTIVE SUMMARY

- **Advances in technology are increasing the demand and diversity of services to maintain infrastructure**
- **Shift in developer preferences to meet the evolving challenges in choices for**
  - Programming languages
  - Database skills
  - Integrated development environments

# INTRODUCTION

## Advances in technology created an ever changing landscape

- Significant impact upon services to support infrastructure

## Maintaining competitive viability

- Report on current trends and identify emerging demands

# METHODOLOGY

- **Data collection methodology**
  - Accessing APIs
  - Scraping websites
- **Perform data wrangling**
  - The two data sets were cleaned, transformed and integrated into a single set for quality control before EDA
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Summarize trends by building a dashboard in Cognos**

# EDA RESULTS

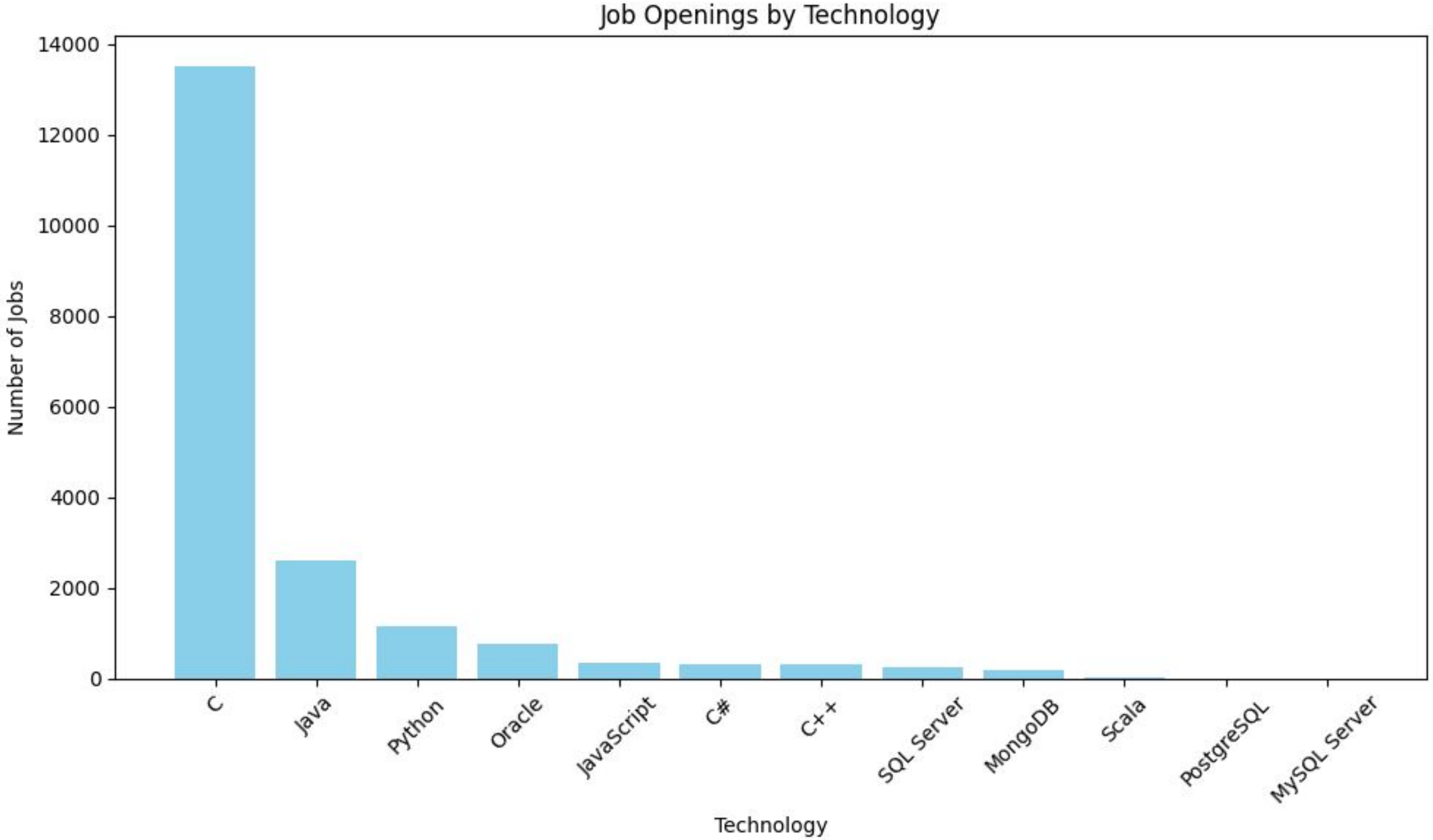
## Current State of IT

- Popular languages
- Employment opportunities
- Compensation

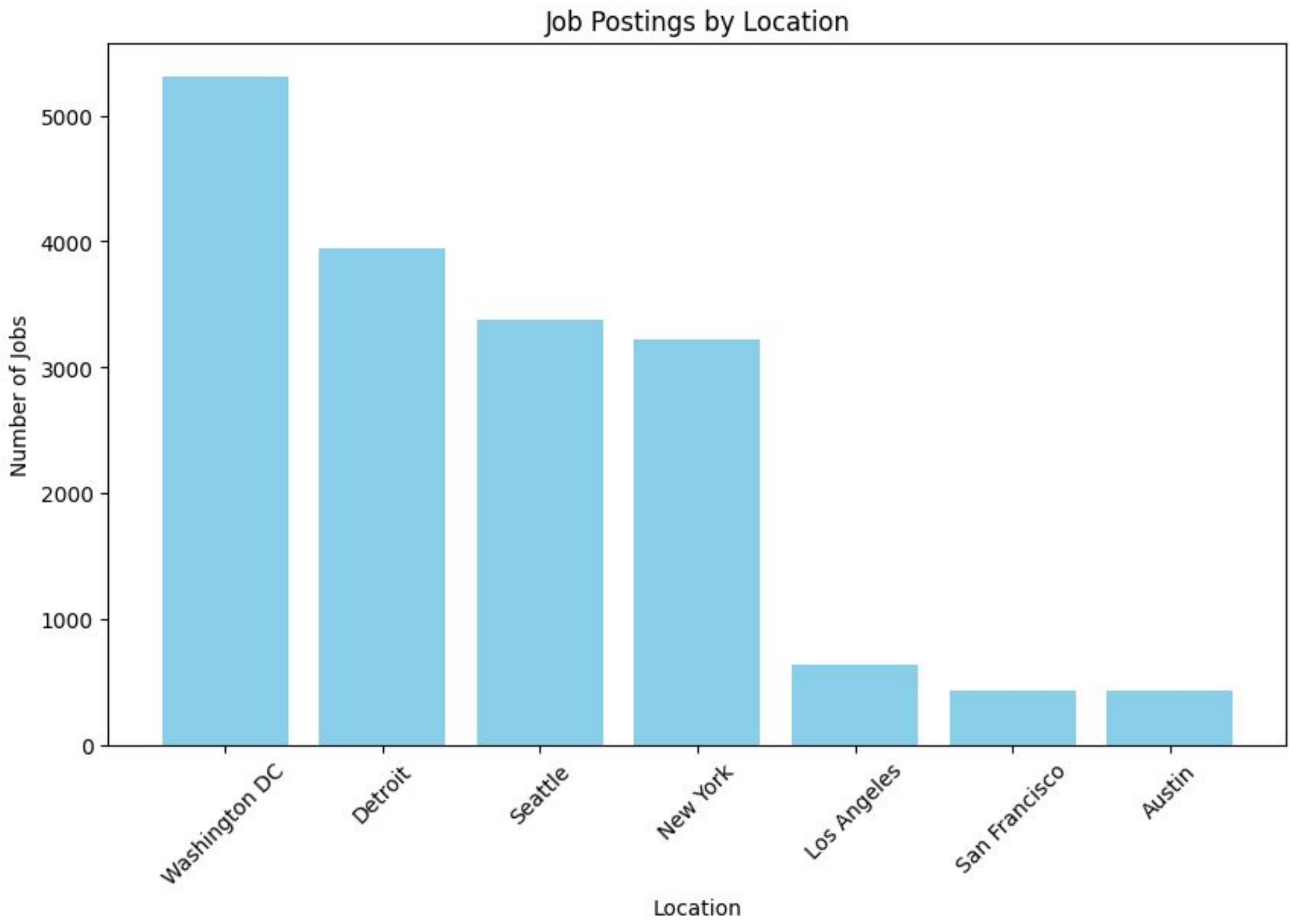
## Emerging Trends

- Programming Languages
- Databases
- Platform
- Web Frame

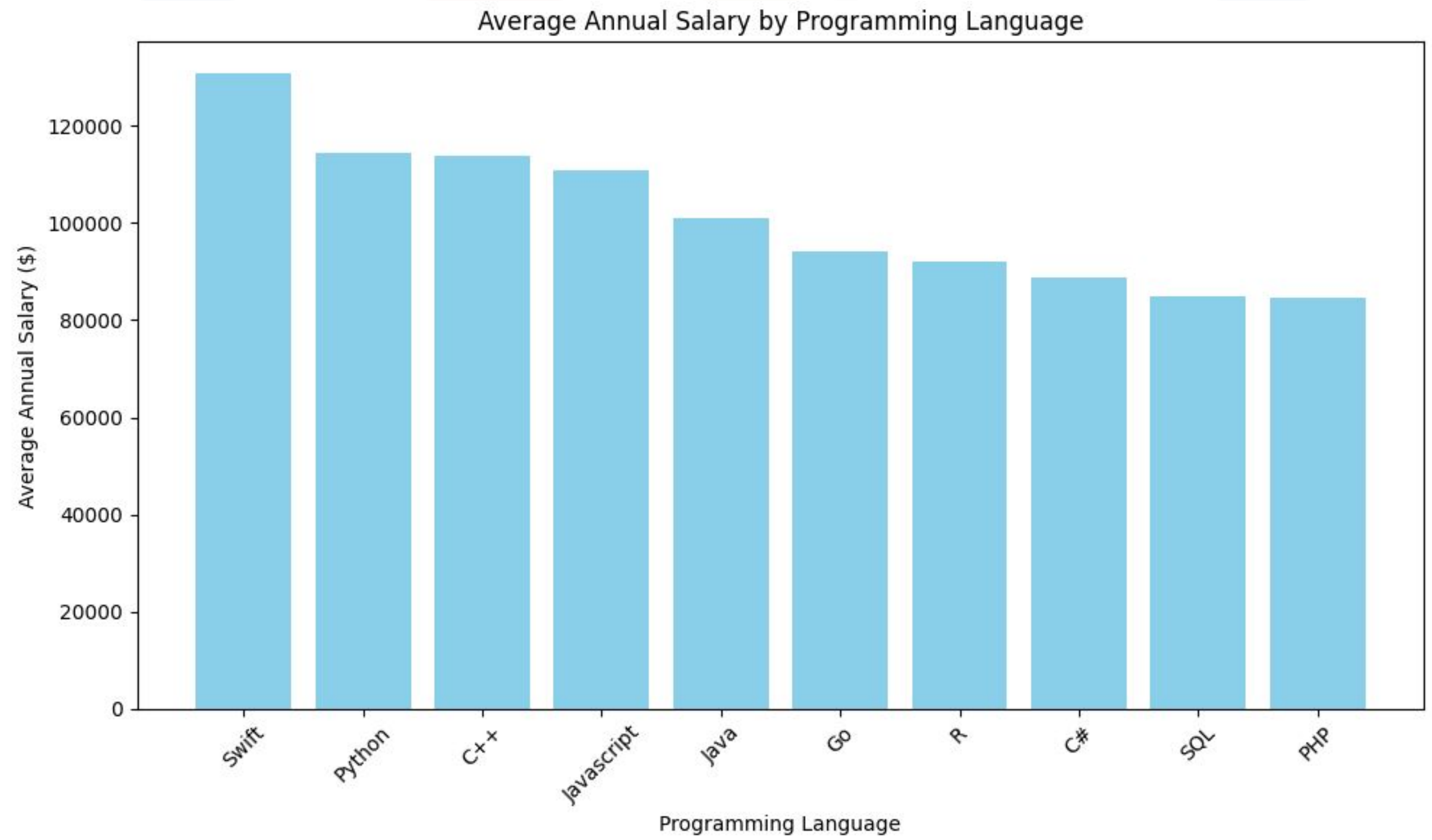
# POPULAR LANGUAGES



# JOB POSTINGS BY LOCATION

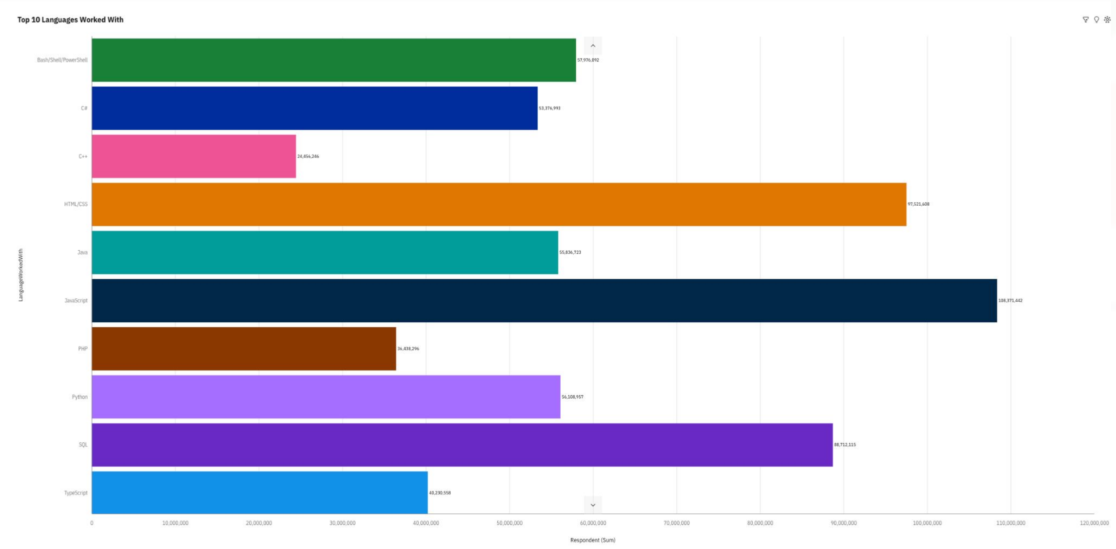


# Annual Salary by Language

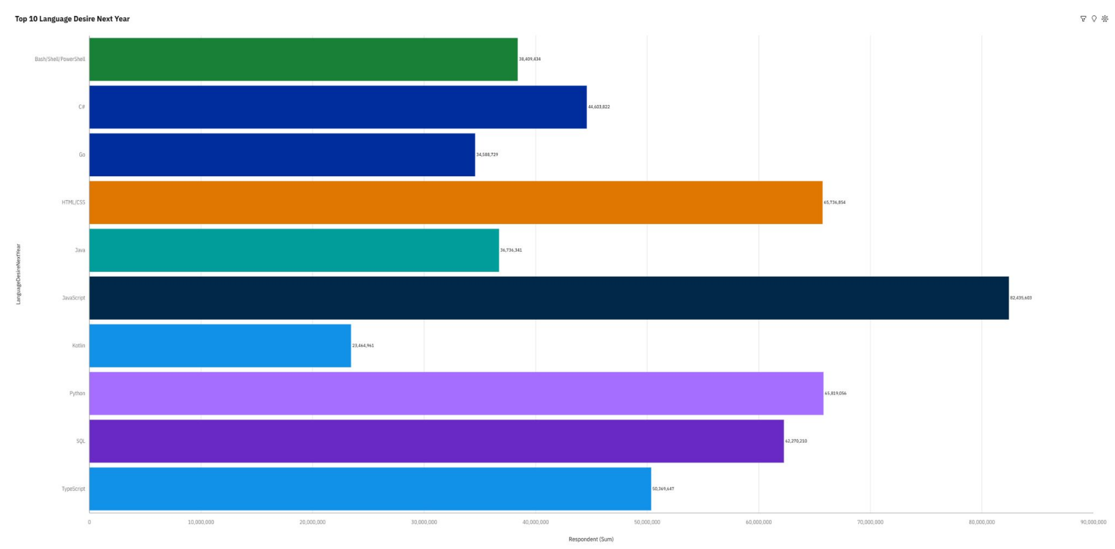


# PROGRAMMING LANGUAGE TRENDS

## Current Year



## Next Year

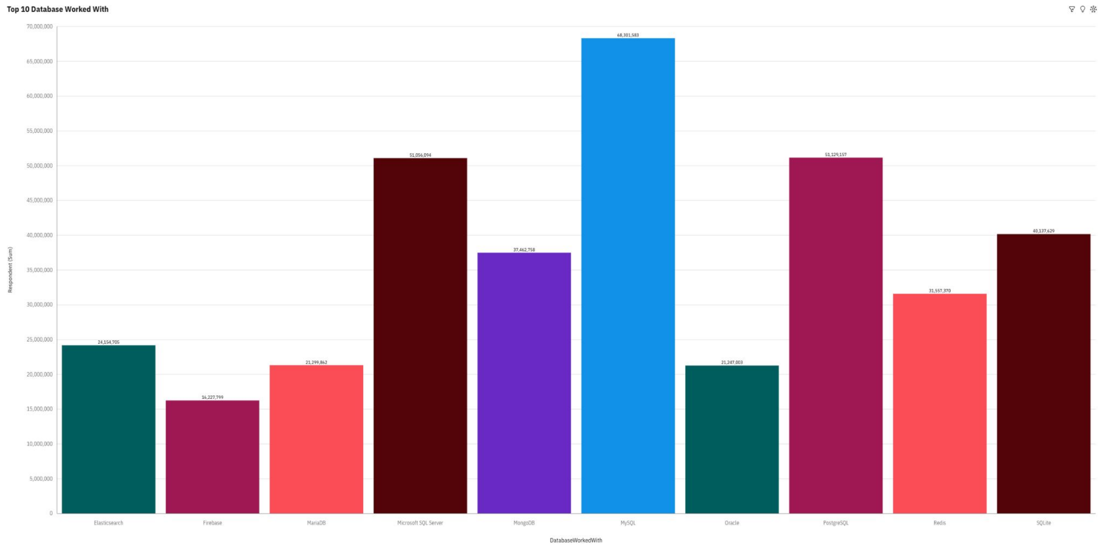


# PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

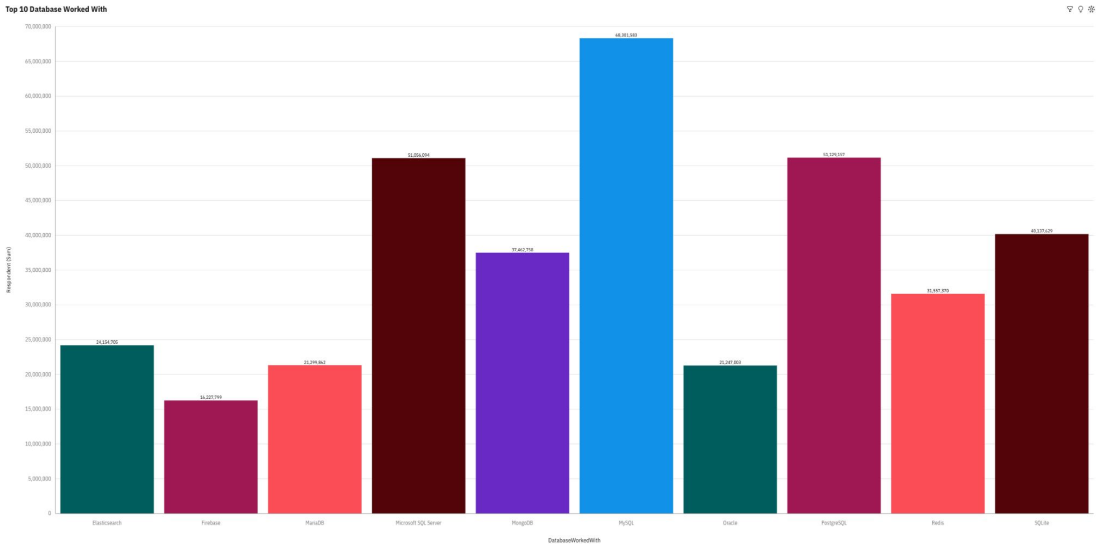
1. **Rise of TypeScript (subset of JavaScript)**
   - Gained significant popularity due to its static typing feature, which helps catch errors early and improves code reliability
   - Projects increasingly prioritizing type safety and maintainability
2. **Continued dominance of JavaScript and Python**
   - Reflecting versatility and widespread use in web development, data science, machine learning, and automation
   - Enduring relevance and adaptability to various programming needs
3. **Steady importance of HTML/CSS and SQL**
   - Essential for web development and database management
   - Consistent ranking underscores the ongoing need for web technologies and data handling skills

# DATABASE TRENDS

## Current Year



## Next Year

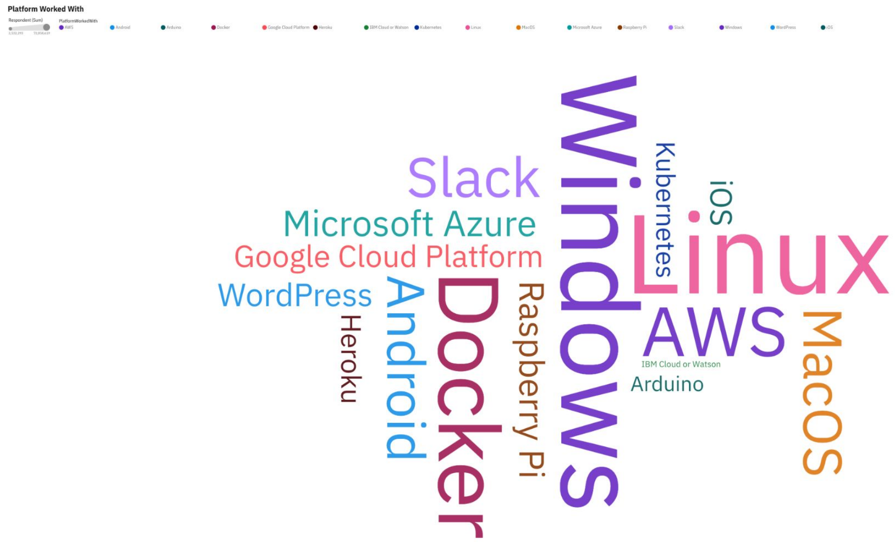


# DATABASE TRENDS - FINDINGS & IMPLICATIONS

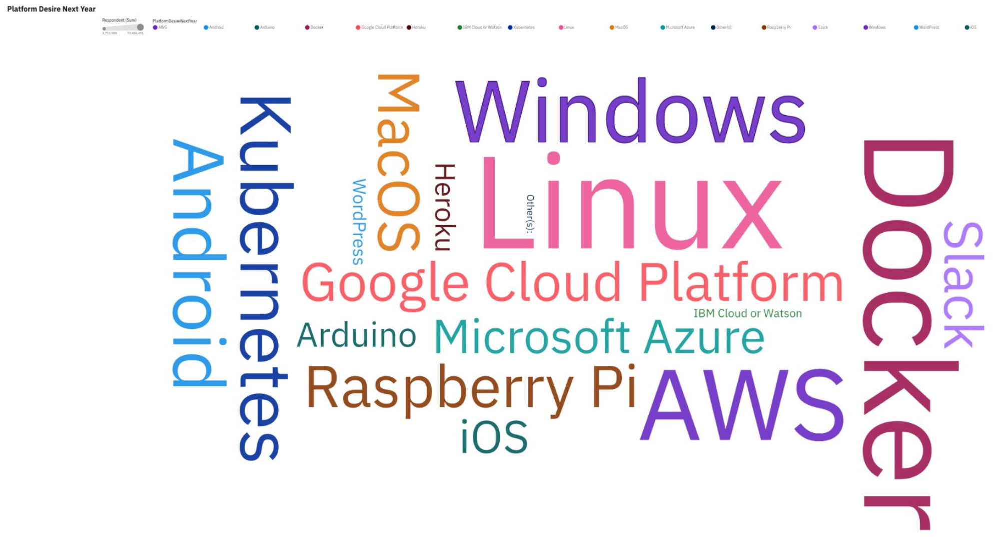
1. **Rise of PostgreSQL**
   - Rise in popularity indicates a strong preference for its advanced features, performance, and open-source nature
   - Go-to choice for many developers and enterprises
2. **Increased Use of MongoDB**
   - Highlights the growing adoption of NoSQL databases for handling large volumes of unstructured data
   - Common in modern applications
3. **Redis Gaining Popularity**
   - In-memory data structure store reflecting the increasing need for high-performance caching and real-time data processing solutions
4. **Steady Use of MySQL and Microsoft SQL**
   - Remain popular choices indicating their reliability and widespread use in various applications and industries

# PLATFORM TRENDS

## Current Year



## Next Year



IBM Developer

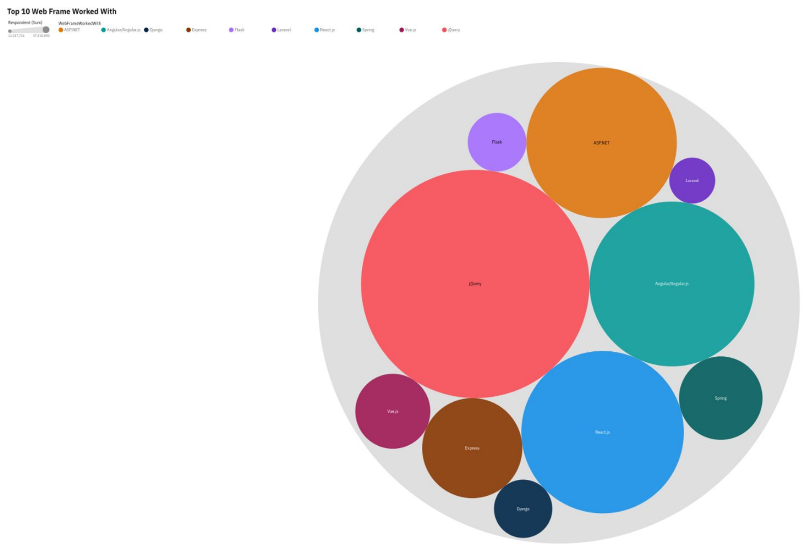
SKILLS NETWORK

# PLATFORM TRENDS - FINDINGS & IMPLICATIONS

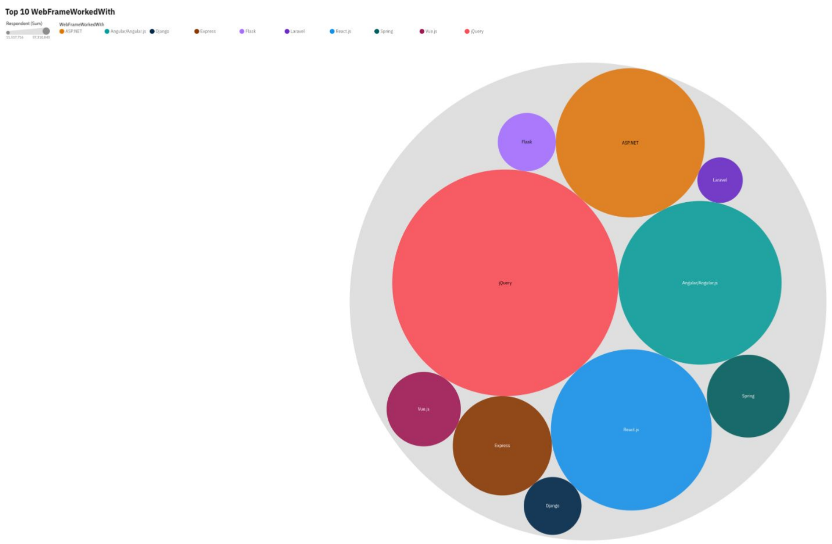
1. **Steady Importance of Linux and Docker**
   - Continue to be highly relevant, reflecting their strong presence in server management, containerization, and development environments
2. **AWS Dominance**
   - A top platform, indicating its ongoing dominance in cloud computing and its critical role in supporting a wide range of applications and services
3. **Windows and Android Rising**
   - Suggests an increasing focus on desktop and mobile operating systems, possibly driven by the growing importance of unified and cross-platform development environments
4. **Shift in Developer Tools and Collaboration**
   - Replacement of Slack with Android might indicate a shift towards mobile-first development and collaboration tools, as well as the increasing importance of mobile platforms in the tech ecosystem

# WEB FRAME TRENDS

## Current Year



## Next Year



# WEB FRAME TRENDS - FINDINGS & IMPLICATIONS

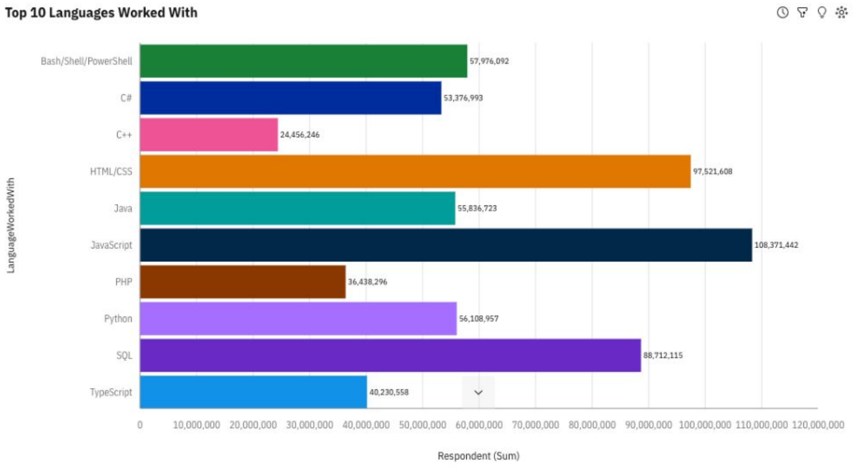
1. **Continued Popularity of React.js**
   - A top framework indicating its ongoing popularity and widespread use for building user interfaces
2. **Rise of Vue.js**
   - Highlights its growing adoption among developers, likely due to its simplicity and ease of integration with other projects
3. **Steady Use of Angular.js**
   - A strong contender, reflecting its robust features and capabilities for building complex applications
4. **Persistence of iQuery**
   - Continued presence suggests that it remains a valuable tool for simplifying JavaScript tasks and DOM manipulation
5. **Emergence of ASP.net**
   - An increasing interest in Microsoft's web development framework, possibly driven by its integration with other Microsoft services and tools

# DASHBOARD

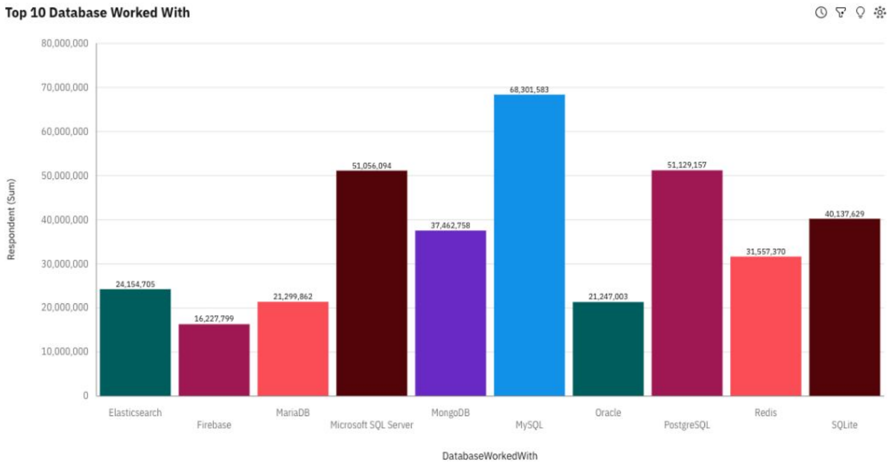
## Cognos Analytics Dashboard

https://github.com/blarghwtfbbq/Capstone_Global_IT/blob/9aef859b9239c297efcc5462943773fdc518fe89/Cognos_Analytics_Dashboard.pdf

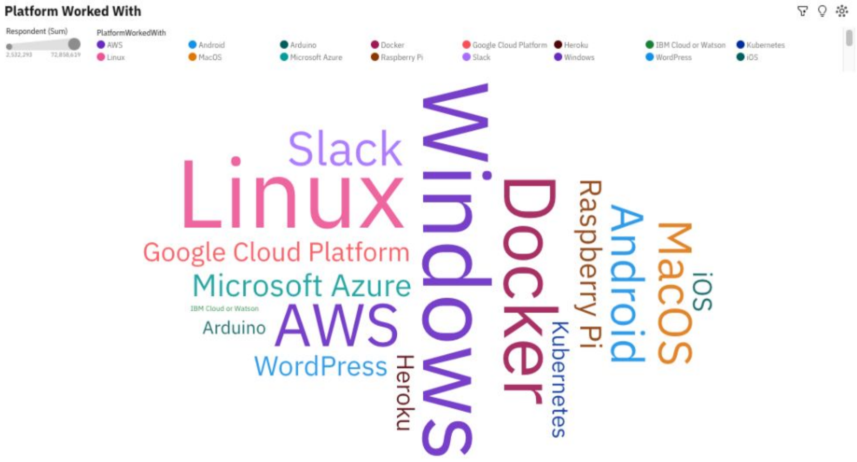
# Current Technology Usage



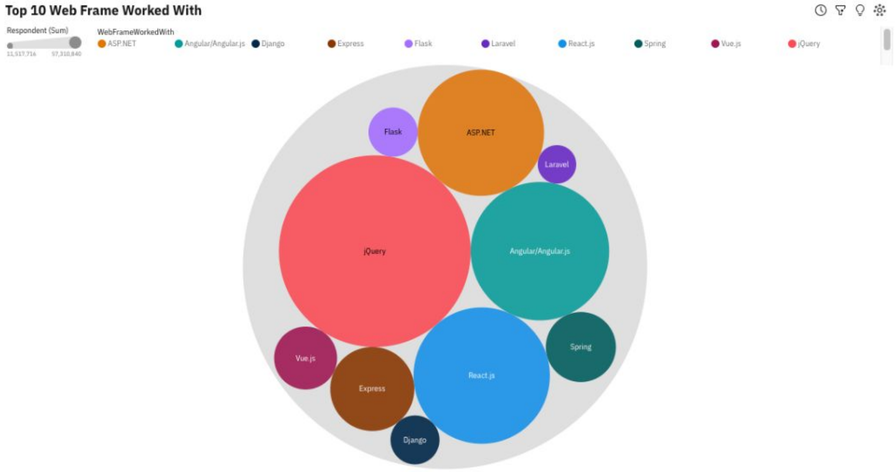
Top 10 Languages Worked With



Top 10 Database Worked With

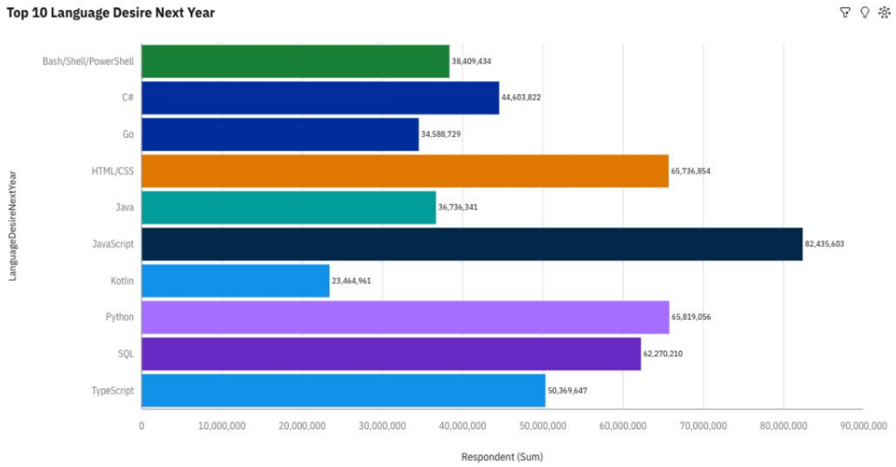


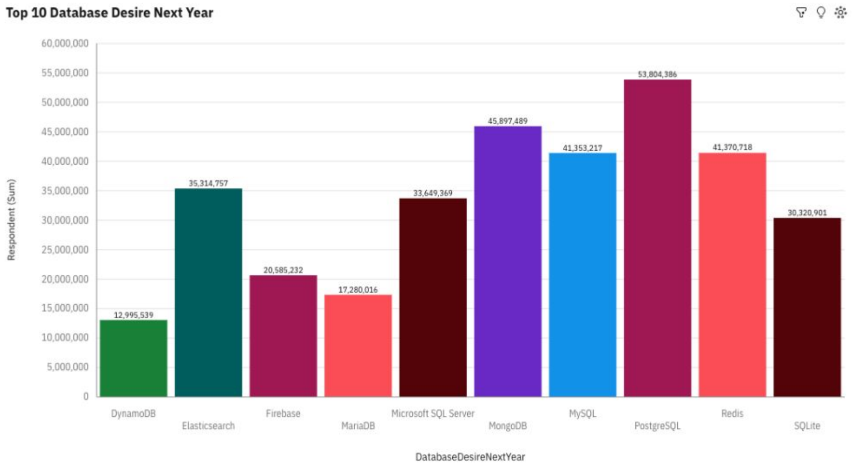
Platform Worked With

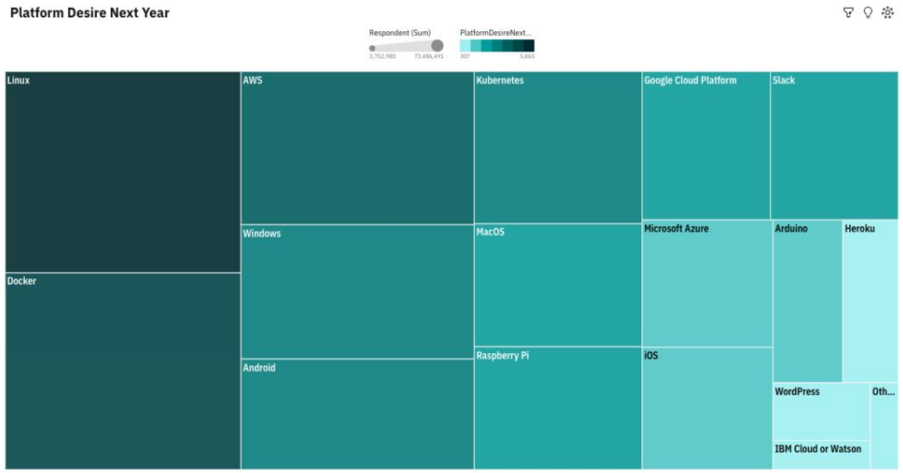


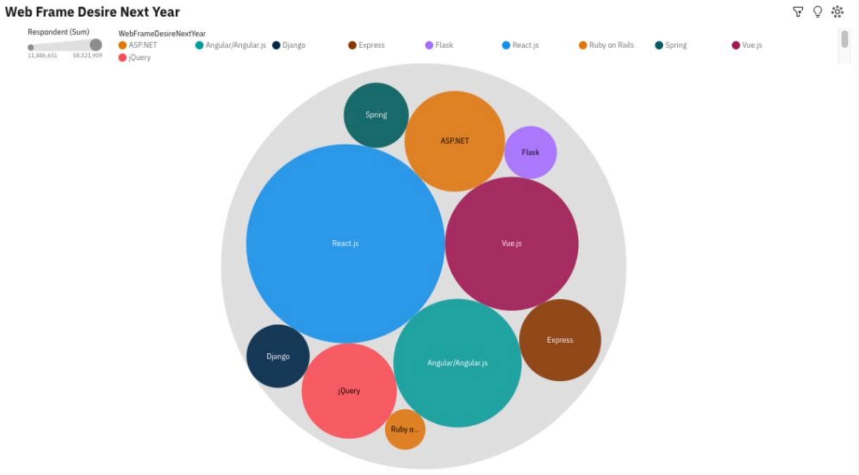
Top 10 Web Frame Worked With

# Future Technology Trend


Top 10 Language Desire Next Year

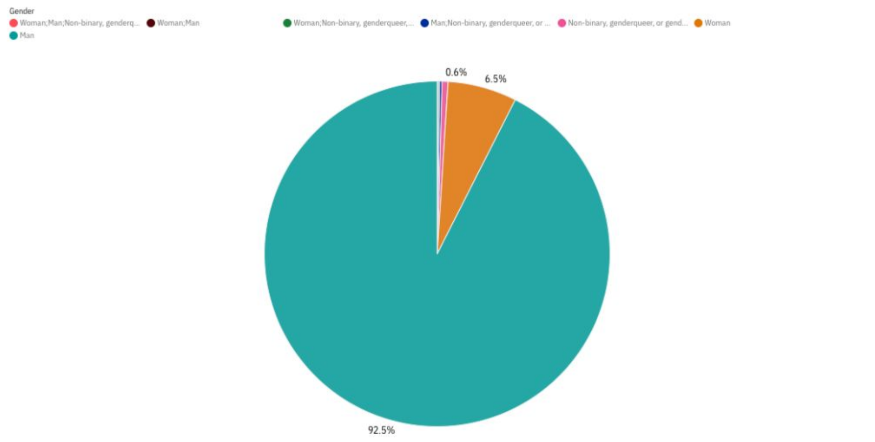

Top 10 Database Desire Next Year


Platform Desire Next Year

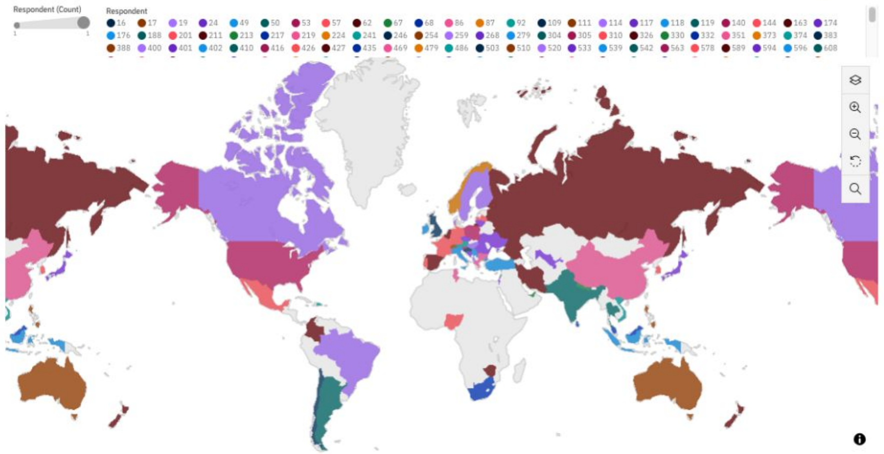

Web Frame Desire Next Year

# Demographics

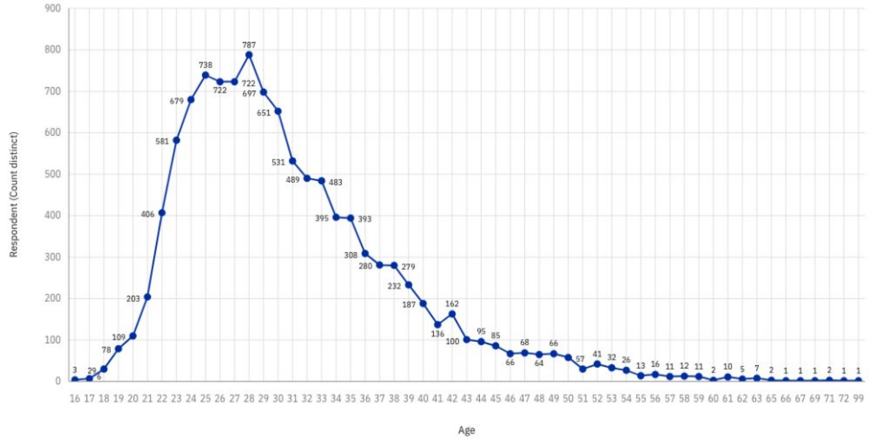
Respondent classified by Gender



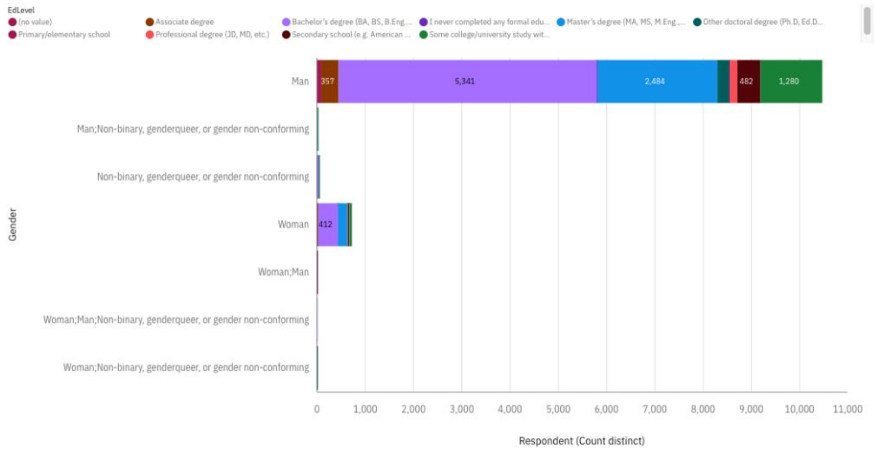
Respondent Count for Countries



Respondent Count by Age



Respondent Count by Gender, classified by Formal Education Level



# DISCUSSION

## Main takeaways

1. Evolving developer preferences in response to industry demands
2. Support must continually evolve to meet new challenges and opportunities in the dynamic nature of the tech industry

# CONCLUSION

1. High tech is a dynamic industry where new technologies and methodologies emerge and gain traction as developers seek better performance, ease of use, and integration capabilities
2. Some of this change is driven by prioritizing more robust and maintainable codebases, with an emphasis on type safety and modern development practices.
3. Through adoption of more diverse and specialized database solutions, it is possible to address different needs such as performance, scalability, and flexibility.
4. Transition to more integrated and versatile platforms that support both development and end-user applications, with a strong emphasis on cloud services and mobile computing.
5. Building a more open-source, in-memory modern and versatile frameworks that cater to different development needs and preferences are becoming more prevalent in current development environments

# APPENDIX

## Supplemental Information

- Github links to course modules
- Code snippets from the different stages of data collections and analysis

# Github Links to Coursera Modules

**Module 1**

- https://github.com/blarghwtfbbq/Capstone_Global_IT/blob/984c6d42e70174c2b61e2e4003293a2d6f251d74/M1Collecting_Jobs_data_Using_API_Questions_Mod_1A_FINAL.ipynb
- https://github.com/blarghwtfbbq/Capstone_Global_IT/blob/984c6d42e70174c2b61e2e4003293a2d6f251d74/M1Web_Scraping_Lab_Mod_1B_FINAL.ipynb
- https://github.com/blarghwtfbbq/Capstone_Global_IT/blob/984c6d42e70174c2b61e2e4003293a2d6f251d74/M1ExploreDataSet_lab_Mod_1C_FINAL.ipynb

**Module 2**

- https://github.com/blarghwtfbbq/Capstone_Global_IT/blob/984c6d42e70174c2b61e2e4003293a2d6f251d74/M2DataWrangling_lab_Mod_2_FINAL.ipynb

**Module 3**

- https://github.com/blarghwtfbbq/Capstone_Global_IT/blob/984c6d42e70174c2b61e2e4003293a2d6f251d74/M3ExploratoryDataAnalysis_lab_Mod_3_FINAL.ipynb

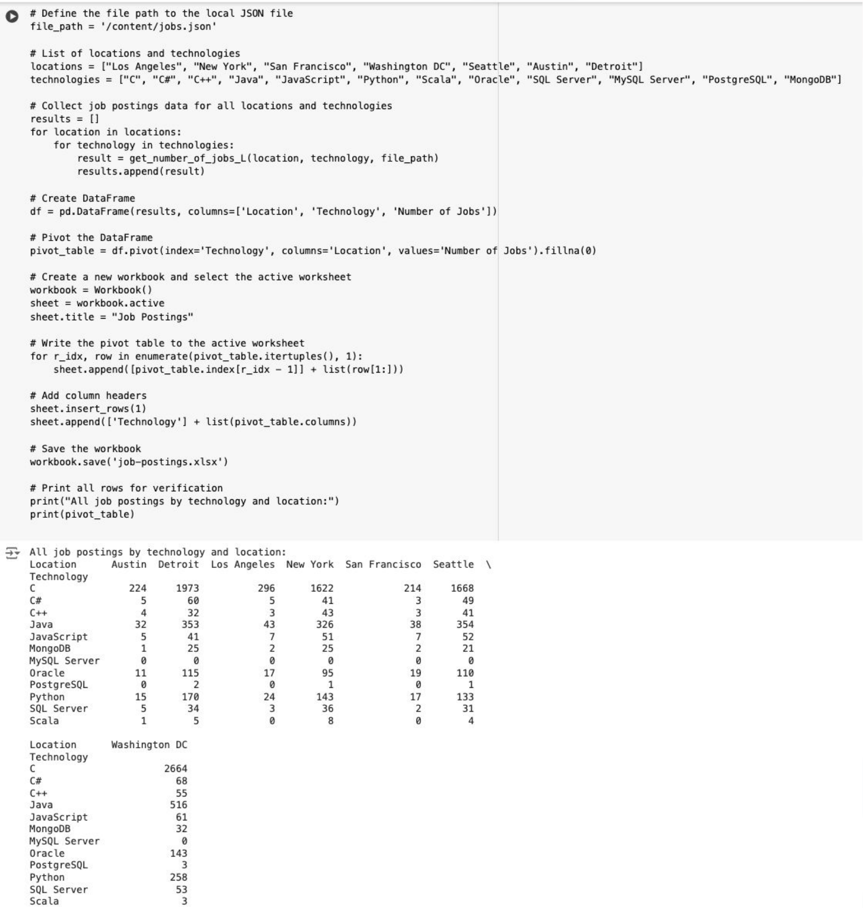
**Module 4**

- https://github.com/blarghwtfbbq/Capstone_Global_IT/blob/984c6d42e70174c2b61e2e4003293a2d6f251d74/M4DataVisualization_lab_Mod_4_FINAL.ipynb

**Module 5**

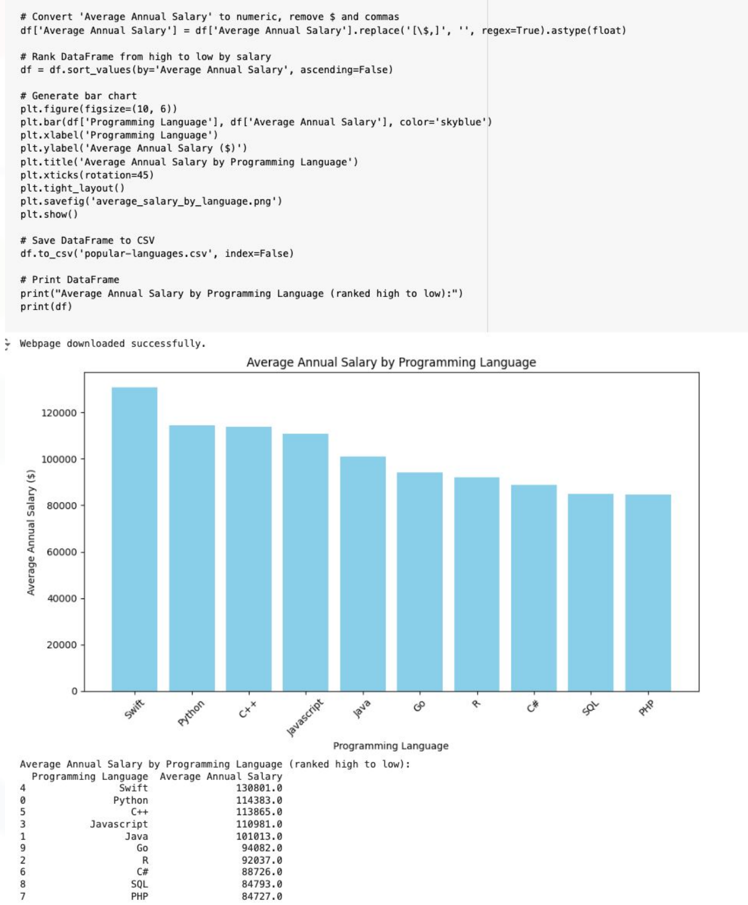
- https://github.com/blarghwtfbbq/Capstone_Global_IT/blob/984c6d42e70174c2b61e2e4003293a2d6f251d74/Cognos_Analytics_Dashboard.pdf

# Data Collection



```
# Define the file path to the local JSON file
file_path = '/content/jobs.json'

# List of locations and technologies
locations = ["Los Angeles", "New York", "San Francisco", "Washington DC", "Seattle", "Austin", "Detroit"]
technologies = ["C", "C#", "C++", "Java", "JavaScript", "Python", "Scala", "Oracle", "SQL Server", "MySQL Server", "PostgreSQL", "MongoDB"]

# Collect job postings data for all locations and technologies
results = []
for location in locations:
    for technology in technologies:
        result = get_number_of_jobs_L(location, technology, file_path)
        results.append(result)

# Create DataFrame
df = pd.DataFrame(results, columns=['Location', 'Technology', 'Number of Jobs'])

# Pivot the DataFrame
pivot_table = df.pivot(index='Technology', columns='Location', values='Number of Jobs').fillna(0)

# Create a new workbook and select the active worksheet
workbook = Workbook()
sheet = workbook.active
sheet.title = "Job Postings"

# Write the pivot table to the active worksheet
for r_idx, row in enumerate(pivot_table.itertuples(), 1):
    sheet.append([pivot_table.index[r_idx - 1]] + list(row[1:]))

# Add column headers
sheet.insert_rows(1)
sheet.append(['Technology'] + list(pivot_table.columns))

# Save the workbook
workbook.save('job-postings.xlsx')

# Print all rows for verification
print("All job postings by technology and location:")
print(pivot_table)
```

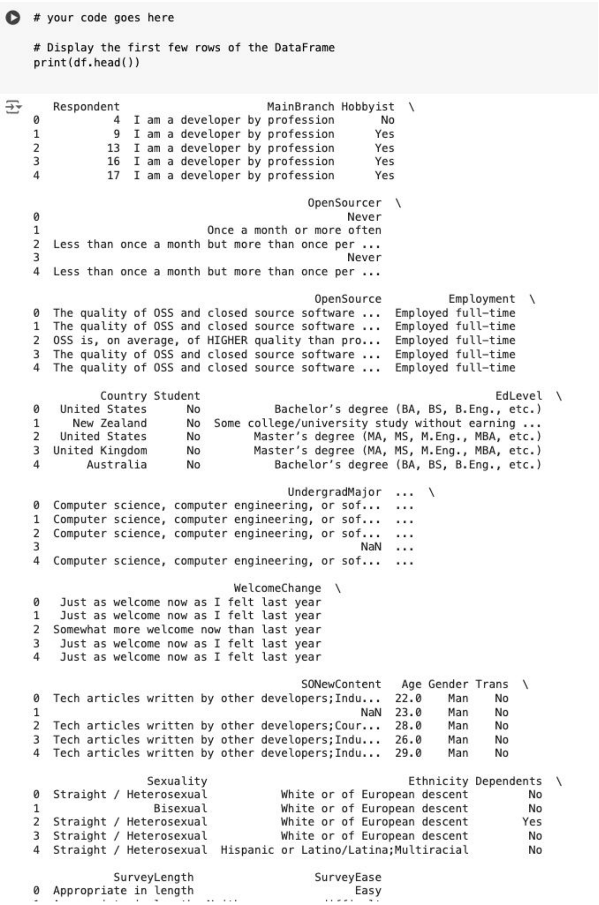
```
All job postings by technology and location:
Location      Austin  Detroit  Los Angeles  New York  San Francisco  Seattle  \
Technology
C                224     1973          296      1622            214     1668
C#                 5       60            5        41              3       49
C++                4       32            3        43              3       41
Java              32      353           43       326             38      354
JavaScript         5       41            7        51              7       52
MongoDB            1       25            2        25              2       21
MySQL Server       0        0            0         0              0        0
Oracle            11      115           17        95             19      110
PostgreSQL         0        2            0         1              0        1
Python            15      170           24       143             17      133
SQL Server         5       34            3        36              2       31
Scala              1        5            0         8              0        4

Location      Washington DC
Technology
C                      2664
C#                       68
C++                      55
Java                    516
JavaScript               61
MongoDB                  32
MySQL Server              0
Oracle                  143
PostgreSQL                3
Python                  258
SQL Server               53
Scala                     3
```



```
# Convert 'Average Annual Salary' to numeric, remove $ and commas
df['Average Annual Salary'] = df['Average Annual Salary'].replace('[\$,]', '', regex=True).astype(float)

# Rank DataFrame from high to low by salary
df = df.sort_values(by='Average Annual Salary', ascending=False)

# Generate bar chart
plt.figure(figsize=(10, 6))
plt.bar(df['Programming Language'], df['Average Annual Salary'], color='skyblue')
plt.xlabel('Programming Language')
plt.ylabel('Average Annual Salary ($)')
plt.title('Average Annual Salary by Programming Language')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('average_salary_by_language.png')
plt.show()

# Save DataFrame to CSV
df.to_csv('popular-languages.csv', index=False)

# Print DataFrame
print("Average Annual Salary by Programming Language (ranked high to low):")
print(df)
```

```
Webpage downloaded successfully.
```

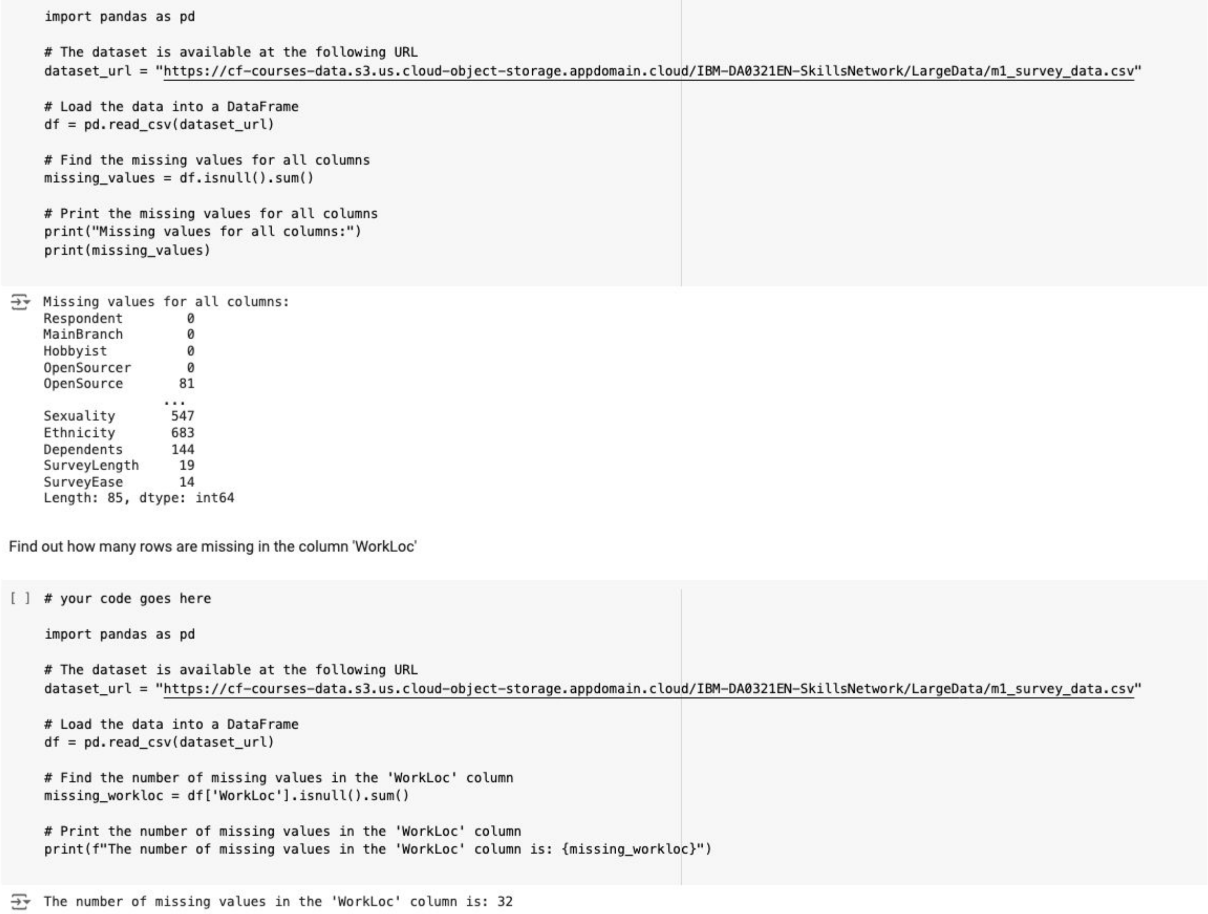
```
Average Annual Salary by Programming Language (ranked high to low):
  Programming Language  Average Annual Salary
4                Swift               130801.0
0               Python               114383.0
5                  C++               113865.0
3           Javascript               110981.0
1                 Java               101013.0
9                   Go                94082.0
2                    R                92037.0
6                   C#                88726.0
8                  SQL                84793.0
7                  PHP                84727.0
```



```
# your code goes here

# Display the first few rows of the DataFrame
print(df.head())
```

```
   Respondent                     MainBranch Hobbyist  \
0           4  I am a developer by profession       No
1           9  I am a developer by profession      Yes
2          13  I am a developer by profession      Yes
3          16  I am a developer by profession      Yes
4          17  I am a developer by profession      Yes

                                          OpenSourcer  \
0                                               Never
1                          Once a month or more often
2  Less than once a month but more than once per ...
3                                               Never
4  Less than once a month but more than once per ...

                                          OpenSource          Employment  \
0  The quality of OSS and closed source software ...  Employed full-time
1  The quality of OSS and closed source software ...  Employed full-time
2  OSS is, on average, of HIGHER quality than pro...  Employed full-time
3  The quality of OSS and closed source software ...  Employed full-time
4  The quality of OSS and closed source software ...  Employed full-time

          Country Student                                            EdLevel  \
0   United States      No           Bachelor’s degree (BA, BS, B.Eng., etc.)
1     New Zealand      No  Some college/university study without earning ...
2   United States      No        Master’s degree (MA, MS, M.Eng., MBA, etc.)
3  United Kingdom      No        Master’s degree (MA, MS, M.Eng., MBA, etc.)
4       Australia      No           Bachelor’s degree (BA, BS, B.Eng., etc.)

                                       UndergradMajor  ...  \
0  Computer science, computer engineering, or sof...  ...
1  Computer science, computer engineering, or sof...  ...
2  Computer science, computer engineering, or sof...  ...
3                                                NaN  ...
4  Computer science, computer engineering, or sof...  ...

                             WelcomeChange  \
0   Just as welcome now as I felt last year
1   Just as welcome now as I felt last year
2  Somewhat more welcome now than last year
3   Just as welcome now as I felt last year
4   Just as welcome now as I felt last year

                                        SONewContent   Age Gender Trans  \
0  Tech articles written by other developers;Indu...  22.0    Man    No
1                                                NaN  23.0    Man    No
2  Tech articles written by other developers;Cour...  28.0    Man    No
3  Tech articles written by other developers;Indu...  26.0    Man    No
4  Tech articles written by other developers;Indu...  29.0    Man    No

              Sexuality                              Ethnicity Dependents  \
0  Straight / Heterosexual          White or of European descent         No
1                Bisexual          White or of European descent         No
2  Straight / Heterosexual          White or of European descent        Yes
3  Straight / Heterosexual          White or of European descent         No
4  Straight / Heterosexual  Hispanic or Latino/Latina;Multiracial         No

           SurveyLength                  SurveyEase
0  Appropriate in length                       Easy
```

# Data Wrangling



```
import pandas as pd

# The dataset is available at the following URL
dataset_url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m1_survey_data.csv"

# Load the data into a DataFrame
df = pd.read_csv(dataset_url)

# Find the missing values for all columns
missing_values = df.isnull().sum()

# Print the missing values for all columns
print("Missing values for all columns:")
print(missing_values)
```

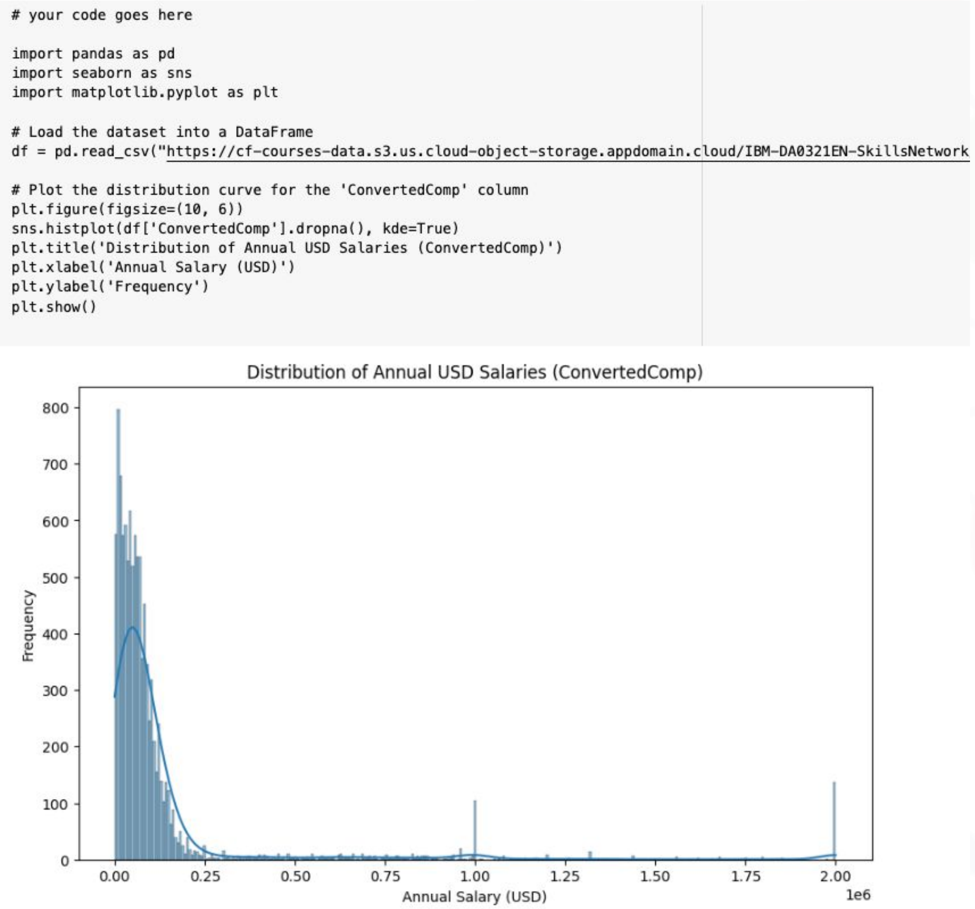
```
Missing values for all columns:
Respondent        0
MainBranch        0
Hobbyist          0
OpenSourcer       0
OpenSource       81
                ...
Sexuality       547
Ethnicity       683
Dependents      144
SurveyLength     19
SurveyEase       14
Length: 85, dtype: int64
```

Find out how many rows are missing in the column 'WorkLoc'

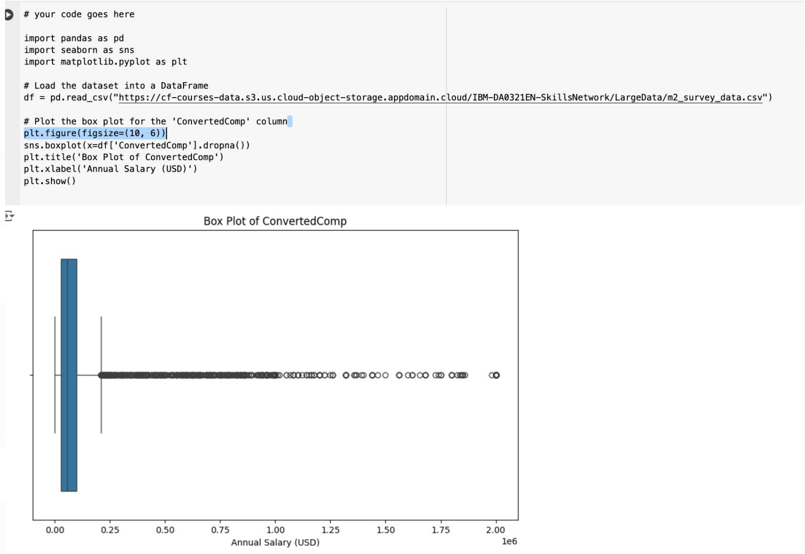
```
[ ] # your code goes here

    import pandas as pd

    # The dataset is available at the following URL
    dataset_url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m1_survey_data.csv"

    # Load the data into a DataFrame
    df = pd.read_csv(dataset_url)

    # Find the number of missing values in the 'WorkLoc' column
    missing_workloc = df['WorkLoc'].isnull().sum()

    # Print the number of missing values in the 'WorkLoc' column
    print(f"The number of missing values in the 'WorkLoc' column is: {missing_workloc}")
```

```
The number of missing values in the 'WorkLoc' column is: 32
```

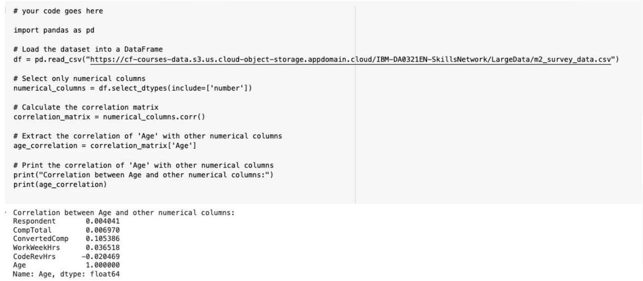
# EDA Visualizations



```
# your code goes here

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset into a DataFrame
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork

# Plot the distribution curve for the 'ConvertedComp' column
plt.figure(figsize=(10, 6))
sns.histplot(df['ConvertedComp'].dropna(), kde=True)
plt.title('Distribution of Annual USD Salaries (ConvertedComp)')
plt.xlabel('Annual Salary (USD)')
plt.ylabel('Frequency')
plt.show()
```



```
# your code goes here

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset into a DataFrame
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m2_survey_data.csv")

# Plot the box plot for the 'ConvertedComp' column
plt.figure(figsize=(10, 6))
sns.boxplot(x=df['ConvertedComp'].dropna())
plt.title('Box Plot of ConvertedComp')
plt.xlabel('Annual Salary (USD)')
plt.show()
```



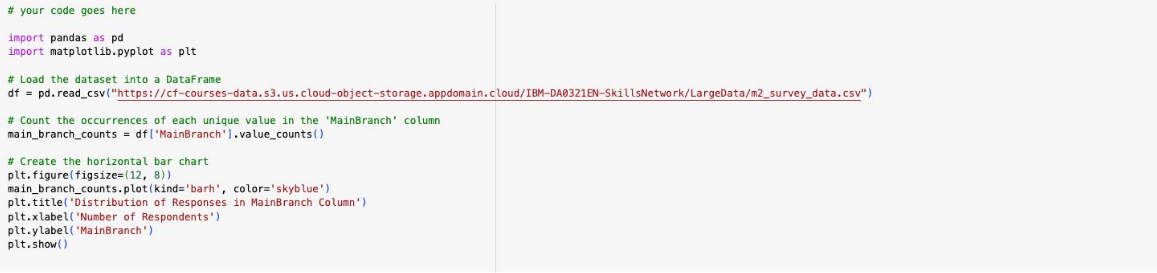
```
# your code goes here

import pandas as pd

# Load the dataset into a DataFrame
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m2_survey_data.csv")

# Select only numerical columns
numerical_columns = df.select_dtypes(include=['number'])

# Calculate the correlation matrix
correlation_matrix = numerical_columns.corr()

# Extract the correlation of 'Age' with other numerical columns
age_correlation = correlation_matrix['Age']

# Print the correlation of 'Age' with other numerical columns
print("Correlation between Age and other numerical columns:")
print(age_correlation)
```

```
Correlation between Age and other numerical columns:
Respondent       0.004041
CompTotal        0.006970
ConvertedComp    0.105386
WorkWeekHrs      0.036518
CodeRevHrs      -0.020469
Age              1.000000
Name: Age, dtype: float64
```

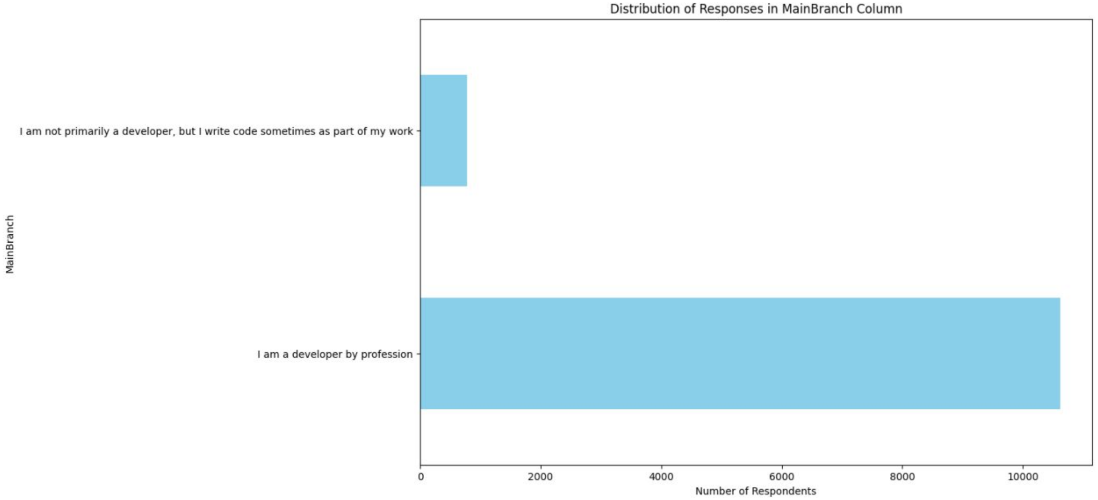
IBM Developer

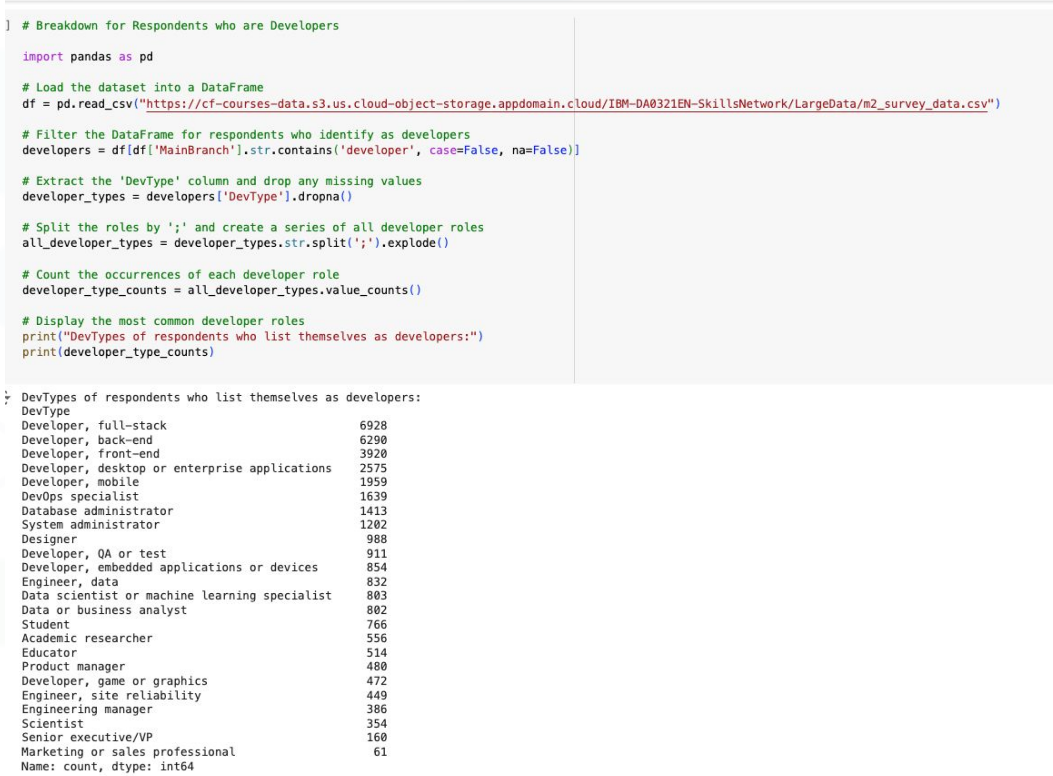
SKILLS NETWORK

# EDA SQL Queries



```python
# your code goes here

import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset into a DataFrame
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m2_survey_data.csv")

# Count the occurrences of each unique value in the 'MainBranch' column
main_branch_counts = df['MainBranch'].value_counts()

# Create the horizontal bar chart
plt.figure(figsize=(12, 8))
main_branch_counts.plot(kind='barh', color='skyblue')
plt.title('Distribution of Responses in MainBranch Column')
plt.xlabel('Number of Respondents')
plt.ylabel('MainBranch')
plt.show()
```





```python
# Breakdown for Respondents who are Developers

import pandas as pd

# Load the dataset into a DataFrame
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m2_survey_data.csv")

# Filter the DataFrame for respondents who identify as developers
developers = df[df['MainBranch'].str.contains('developer', case=False, na=False)]

# Extract the 'DevType' column and drop any missing values
developer_types = developers['DevType'].dropna()

# Split the roles by ';' and create a series of all developer roles
all_developer_types = developer_types.str.split(';').explode()

# Count the occurrences of each developer role
developer_type_counts = all_developer_types.value_counts()

# Display the most common developer roles
print("DevTypes of respondents who list themselves as developers:")
print(developer_type_counts)
```

```
DevTypes of respondents who list themselves as developers:
DevType
Developer, full-stack                           6928
Developer, back-end                             6290
Developer, front-end                            3920
Developer, desktop or enterprise applications   2575
Developer, mobile                               1959
DevOps specialist                               1639
Database administrator                          1413
System administrator                            1202
Designer                                         988
Developer, QA or test                            911
Developer, embedded applications or devices      854
Engineer, data                                   832
Data scientist or machine learning specialist    803
Data or business analyst                         802
Student                                          766
Academic researcher                              556
Educator                                         514
Product manager                                  480
Developer, game or graphics                      472
Engineer, site reliability                       449
Engineering manager                              386
Scientist                                        354
Senior executive/VP                              160
Marketing or sales professional                   61
Name: count, dtype: int64
```